

Mixed signal SIMD processor array vision chip for real-time image processing

Stephen J. Carey · David R. W. Barr ·
Bin Wang · Alexey Lopich · Piotr Dudek

Received: 10 June 2013 / Revised: 16 September 2013 / Accepted: 23 September 2013 / Published online: 31 October 2013
© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract A prototype vision chip has been designed that incorporates a 20×64 array of processing elements on a $31 \mu\text{m}$ pitch. Each processor element includes 14 bits of digital memory in addition to seven analogue registers. Digital operands include NOR and NOT with operations of diffusion, subtraction, inversion and squaring available in the analogue domain. The cells of the array can be configured as an asynchronous propagation network allowing operations such as flood filling to occur with times of $\sim 1 \mu\text{s}$ across the array. Exploiting this feature allows the chip to recognise the difference between closed and open shapes at 30,000 frames per second. The chip is fabricated in $0.18 \mu\text{m}$ CMOS technology.

Keywords Vision chip · Smart sensor · Asynchronous image processing · Cellular processor array

1 Introduction

Vision chips integrate image sensing and pixel-level processing on a single silicon die (Fig. 1) and permit massively parallel processing of pixel derived data. Under different configurations, a vision chip can be made to execute image processing algorithms with very low power consumption or, alternatively, operate at very high speeds.

Both of these positive attributes arise from the execution of the algorithm upon an array of processing elements (PEs) on the focal plane. Information is processed within a PE and between near neighbour PEs, with only data extracted from the images communicated between the chip and the system controller. Overall, a reduced dataflow (relative to conventional systems) is accomplished in respect of both low power and high-speed applications. Hence, vision chip based systems have significant promise to yield processed information with superior performance over that of traditional image processing methods that separate the subsystems of imaging and processing. Applications requiring high speed operation have traditionally used ultra-high frame rate cameras [1, 2] coupled to FPGA or PC hardware; while recently some non-conventional approaches [3] have been proposed, all these systems produce prodigious data rates. For such applications, vision chips can offer an alternative solution, that eliminates the sensory readout bottleneck [4–6]. For systems requiring very low power operation, the need to run an ADC merely to establish that there has been no change to the imaged scene, is a distinct disadvantage. The vision chip approach can circumvent the high data-rate issue and obviate the need for continuously operating ADCs to achieve very low power system operation [7].

Recent vision chips, in common with the chip described here, operate as SIMD computational devices; such devices have been presented widely in both digital [8–10], analogue [11–14] and mixed signal form [15, 16]. Designers of wholly digital vision chips have tended to adopt larger cell sizes ($65 \times 25 \mu\text{m}$ [8], $67 \times 64 \mu\text{m}$ [9] and $51 \times 54 \mu\text{m}$ [10]) than their analogue ($49 \times 49 \mu\text{m}$ [11], $35 \times 35 \mu\text{m}$ [12], $34 \times 29 \mu\text{m}$ [13], $26 \times 26 \mu\text{m}$ [14]) counterparts. With smaller semiconductor process nodes (e.g. 45 nm) becoming available, it will be possible to shrink the same

S. J. Carey (✉) · D. R. W. Barr · B. Wang · A. Lopich ·
P. Dudek
School of Electrical and Electronic Engineering, The University
of Manchester, Manchester M13 9PL, UK
e-mail: s.carey@manchester.ac.uk

P. Dudek
e-mail: p.dudek@manchester.ac.uk

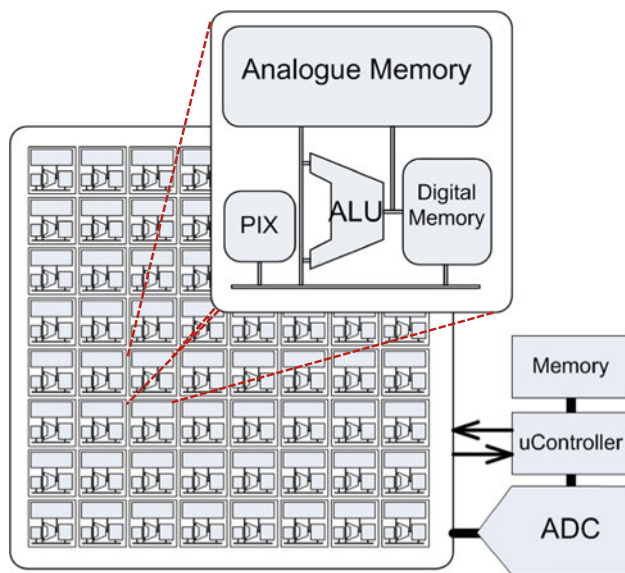


Fig. 1 Vision chip processing architecture—each pixel contains a photodiode (PIX) and a PE (ALU and memory)

digital functionality onto a much smaller silicon area than the analogue equivalent. However, while utilizing lower cost processes, specifically 0.18 μm , analogue processors can achieve bit-equivalent memories and ALUs similar to their digital counterparts in a more compact space. The relatively high pixel pitch of all recent vision chips [8–16] limits the pixel count to typically <20 kpixels, making them suitable for applications in which high speed or low power is required, but very high spatial resolution is not essential, for example, surveillance systems [7, 17], performing on-line analysis in industrial production systems [4], or providing vision systems for robots [18]. The separation of processing and imaging areas (described in [8]) removes some of the constraints on occupied PE silicon space resulting in low fill factor; the design compromise made for this approach is that algorithms must start by reading out data from the array into the close-connected PEs; for fast image processing, frame rates might be slowed by this process.

This paper, which discusses a prototype vision chip designed with a 64×20 pixel array, extends the descriptions provided in [19]. This test IC is intended to be the precursor of a larger array. In Sect. 2, we discuss the architecture of the chip with Sect. 3 describing the circuits incorporated into the PE. Section 4 discusses the programming model and language used to define the operation of the IC, and Sect. 5, the implementation and measurement results. Section 6 relates an example application of the IC. We include a description of a program that distinguishes between open and closed shapes at 30,000 fps, demonstrating the utility of focal-plane processing. With the processor-per-pixel architecture, the program is expected to scale to larger

processor arrays without significant reduction in execution speed. As earlier applications have also demonstrated [20], frames are not read-out at this speed, only the results of processing operations are transferred to the program controller.

2 Architecture overview

The overview of the chip architecture is shown in Fig. 2. In this IC, each PE of the array incorporates (in addition to a photodiode) a mixed set of analogue and digital storage registers, with processing capability in both domains [21] allowing efficient processing and storage of data dependant on data-type. While the chip utilises analogue processing, each PE is a software-programmable sampled data system (hybrid analogue/digital) [22] in common with digital processors rather than a continuous time processor.

The IC is controlled by an external instruction issuing device. All instructions provided to the IC are executed according to the SIMD paradigm, that is, all the processors of the array execute simultaneously the same instruction but upon their own local data. The local data available extends to that of their nearest orthogonal neighbours. The

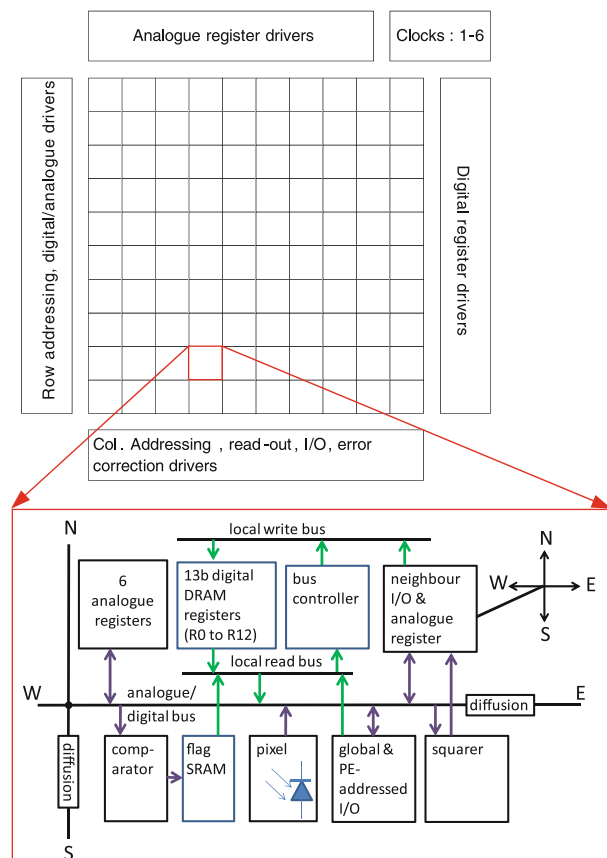


Fig. 2 Architecture of the IC

instruction set supports arithmetic operations of addition, subtraction, negation, squaring and comparison performed on analogue registers, and logic OR and NOR operations performed on binary registers. Hardware that facilitates SIMD programming is a good fit with low-level pixel processing operations, whereby operations are repeated on a per-pixel basis.

The IC described has an architecture that is intended to allow its deployment in many different applications, while the physical pitch of the PEs allows an IC with greater resolution at lower cost to be fabricated in comparison to its predecessor SCAMP-3 [11]—and additionally allows use of smaller optics. The IC utilises switched current memories for analogue storage, with binary storage provided by 3-transistor digital DRAM memories as advanced in ASPA vision chips [10]. The sub-systems that could be incorporated in the PEs of the array were limited by the connectivity that could be provided to the PEs. However, relative to SCAMP-3 with three metal layers available, the six metal layers of the 0.18 μm process has afforded considerable additional features and functionality. The additional metal layers however, also have deleterious effects on sensing uniformity since the photodiode is now at the base of the metal-layer stack; this will partially shadow the light sensitive area. Applications of vision chips such as motion, edge detection, tracking and shape detection require a combination of analogue and digital operations. In the design of this IC, we have shifted the ratio of analogue to digital memory, reducing the number of space-consuming analogue registers while increasing the digital resources available. Furthermore, the digital memories can be used to configure an asynchronous propagation network [23], allowing global operations on data, in addition to performing logical local operations. Also added to this chip is a controllable diffusion network allowing large-scale spatial low-pass filters to occur in a single instruction, and a squarer offering one-quadrant operational capability. The IC provides for global analogue and digital data write and read; PE-specific digital write is also provided, hence allowing PE-specific analogue write via a flagged global analogue write operation.

PEs are controlled by means of a 61-bit instruction code word (ICW), with additional control supplied from 12 analogue inputs. The long ICW is registered on chip and input by means of a 16-bit bus, allowing a reduction in external pin-count. To deliver the ICW to the PE array requires large scale connectivity (relative to the compact PE size) with row and column drivers spread across all four sides of the IC.

The chip supports random access readout of analogue or digital data from the PEs, allowing for full array or region-of-interest readout. Additionally, flexible row and column decoders allow simultaneous addressing of pixel-blocks:

analogue data to be readout is summed to provide regional or global analogue outputs; logic OR operation is performed on the digital output data within the addressed block.

3 Circuits

3.1 Analogue processor fundamentals

The analogue processor in this work is based on the ideas developed in [22, 24]; these have been recently explained in [25] but are included here for completeness. The processor used in this work exploits switched-current (SI) circuit techniques [26]. A basic SI memory cell is illustrated in Fig. 3. When a MOS transistor works in the saturation region its drain current I_{ds} can be, in a first-order approximation, described by:

$$I_{ds} = K(V_{gs} - V_t)^2 \quad (1)$$

where K is the transconductance factor, V_t is the threshold voltage and V_{gs} is the gate-source voltage.

The SI memory cell remembers the value of the input current by storing charge on the gate capacitance C_{gs} of the MOS transistor. When writing to the cell both switches, S and W, are closed (Fig. 3b). The transistor is diode-connected and the input current i_{in} forces the gate-source voltage V_{gs} of the transistor to the value corresponding to the drain current $I_{ds} = I_{REF} + i_{in}$, according to (1). At the end of the write phase the switch W is opened and thus the gate of the transistor is disconnected, i.e. put into a high impedance state. Due to charge conservation on the capacitor C_{gs} , the voltage at the gate V_{gs} will remain constant.

When reading from the memory cell, switch S is closed while switch W remains open (Fig. 3c). Now, the transistor acts as a current source. As the gate-source voltage V_{gs} is the same as one that was set during the write phase, the drain current I_{ds} has to be the same (provided that the transistor remains in the saturation region), and hence the output current i_{out} is equal to

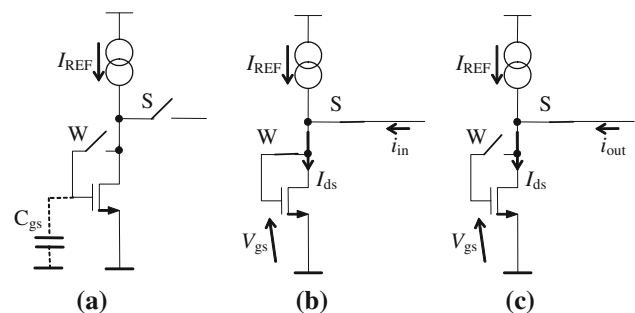


Fig. 3 Basic SI memory cell. **a** During storage both switches are open, **b** both switches are closed when the cell is ‘written to’, **c** only switch ‘S’ is closed when the cell is ‘read from’

$$i_{out} = I_{ds} - I_{REF} = i_{in} \quad (2)$$

Therefore, the SI memory cell is, in principle, capable of storing a continuous-valued (i.e. real) number, within some operational dynamic range.

In addition to data storage, the PE needs to provide a set of basic arithmetic operations. These can be achieved with a very low area overhead in the current-mode system. Consider a system consisting of a number of *registers* implemented as SI memory cells, connected to a common *analogue bus*. Each memory cell can be configured (using corresponding switches S and W) to be read from, or written to (Fig. 4).

The basic operation is the transfer operation, as shown in Fig. 4(a). Register A (configured for reading) provides the current i_A to the analogue bus. This current is consumed by register B (configured for writing); register C is not selected, and hence it is disconnected from the bus. Therefore, the analogue (current) data value is transferred from A to B. This transfer is denoted as $\mathbf{B} \leftarrow \mathbf{A}$. According to the current memory operation, Register B will produce the same current when it is read from (at some later instruction cycle). If we consider that the data value is always the one provided *into the analogue bus*, it can be easily seen that $i_B = -i_A$, i.e. the basic transfer operation includes negation of the stored data value.

The addition operation (and in general current summation of a number of register values) can be achieved by configuring one register for writing, and two (or more) registers for reading. The currents are summed, according to Kirchhoff's current law, directly on the analogue bus. For example, the situation shown in Fig. 4(b) produces the operation $\mathbf{A} \leftarrow \mathbf{B} + \mathbf{C}$. Subtraction is performed by negation followed by addition.

The division operation is achieved by configuring one register cell for reading and many (typically two) registers for writing. The current is split, and if the registers are identical then it is divided equally between the registers that are configured for writing, producing a division by a fixed factor. For example, in Fig. 4(c) both registers A and B will store current equal to half of the current provided by register C. We denote this instruction as $\mathbf{DIV}(\mathbf{A} + \mathbf{B}) \leftarrow \mathbf{C}$.

Multiplication and division by constant factors (e.g. coefficients in a convolution kernel) can be simply achieved by multiple application of add or divide by two operations.

In the presented design, a compact current-mode squarer sub-system [27] is implemented. It provides the squared analogue current value (this is useful for algorithms requiring, for example, energy or mean-squared error calculations), and allows multiplication to be achieved through multiple steps, by determining $(x + y)^2 - x^2 - y^2 (=2xy)$. Other instructions could be implemented in hardware using current mode circuits, however, in a vision chip application silicon area is at a premium, so only those computation elements that can be implemented compactly can be used. Consequently, more silicon area can be devoted to register circuits, increasing the amount of local memory (or improving the accuracy of computations, which largely depends on device matching, i.e. device area). While this is not a one-step command, the squarer requires few transistors and can be fitted into a compact space within the PE.

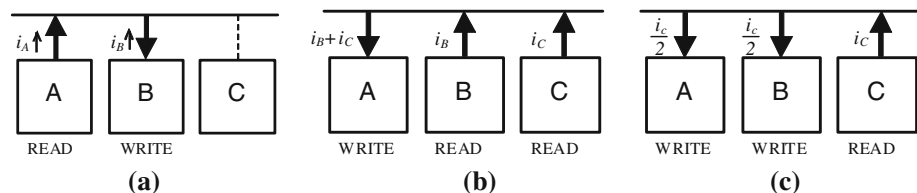
3.2 Analogue sub-system

The analogue sub-system of the PE implemented on the chip comprises six general purpose analogue registers (A,B,C,D,E,F), 1 special purpose and neighbour communication register (“NEWS”), a variable strength diffusion network, a voltage controlled current source (the “IN” register) that facilitates the global input of data to the array, a current squarer, and a photodiode (two types, N-well diode and N+/P-subst, are explored—split equally across the array). Additionally, an analogue comparator detects the sign of a register's stored current, storing the result in a local activity register (1-bit SRAM flag). The 1-bit SRAM flag is used to allow conditional code execution. The flag data can be also transferred to one of the 13 bits of DRAM memory and processed in the digital sub-system. The digital sub-system can also write to the flag register, providing a communication path from digital to analogue domain. The PE includes connectivity to allow either analogue or digital sub-systems to access a common analogue/digital bus allowing exchange of data within the cell or to its orthogonal neighbours.

3.2.1 Analogue memories

The analogue memories of this IC are based on the S²I cell [28]. The standard cell was modified to include an error correction circuit [24] to allow a reduction in cell size

Fig. 4 Basic operations in a current-mode processor:
a register data transfer,
b addition, c division by two



while maintaining low signal-dependent error (since signal-independent errors can be accommodated algorithmically [22]). These errors have their origin in switch signal clock feedthrough, charge injection and output conductance errors; the addition of this circuit allows these errors to be compensated for.

3.2.2 Squarer

The squarer (Fig. 5) is based on the design presented in [27]. It enables 1-quadrant squaring operations within the PE; it reads a current I_{in} from the analogue bus, with one or more general purpose registers sourcing current to the bus. In the design presented below, we use two separate N-wells for the three identical P-MOS transistors, with MP1 and MP2 sharing a different N-well to MP3 (the N-wells are all biased to their respective P-MOS source nodes). The output current, I_{sq} , is stored in the special-purpose NEWS register.

From examination of Fig. 5, with $I_{MP1} = I_{MP2}$ and $I_{MN1} = I_{MN2}$, the following circuit equations can be written:

$$I_{MP1} - I_{MP3} = I_{in} \quad (3)$$

$$I_{sq} = I_{MP3} + I_{MP1} - I_{Msq} \quad (4)$$

$$I_{MP1} = \beta_P (V_{dd} - V_{g1} - V_{tp})^2 \quad (5)$$

$$I_{MP3} = \beta_P (V_{g1} - V_{b1} - V_{tp})^2 \quad (6)$$

After some algebra, from Eqs. (3), (5) and (6) and with the substitution $a = V_{dd} - V_{b1} - 2V_{tp}$ we can derive an expression for current I_{MP1}

$$I_{MP1} = \frac{I_{in}^2}{4a^2\beta_P} + \frac{I_{in}}{2} + \frac{\beta_P a^2}{4} \quad (7)$$

Summing Eqs. (3) and (4) and using the expression for I_{MP1} from Eq. (7):

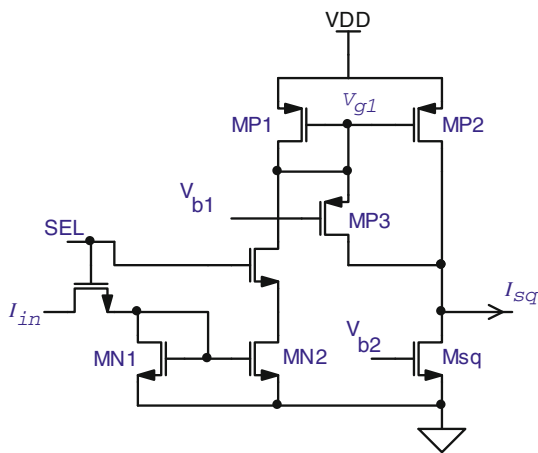


Fig. 5 Squarer schematic

$$I_{sq} = -I_{Msq} + \frac{\beta_P a^2}{2} + \frac{I_{in}^2}{2\beta_P a^2} \quad (8)$$

The squarers of the array are normally set-up via the two global bias voltages, V_{b1} and V_{b2} such that there is no offset current and the first two terms of Eq. (8) sum to zero. When this condition is met, Eq. (8) reduces to:

$$I_{sq} = \frac{I_{in}^2}{4I_{Msq}} \quad (9)$$

The prerequisite of correct circuit operation requires that transistors remain in saturation, with transistor MP3 being the limiting device. This requires that $V_{g1} - V_{b1} \geq V_{tp}$. With substitutions from Eqs. (5) and (7)

$$I_{in} \leq \beta_P a^2 \quad (10)$$

Coupled with the constraint of “no-offset” from Eq. (6), we then have

$$I_{in} \leq 2I_{Msq} \quad (11)$$

In order to maximize the output dynamic range of I_{sq} , I_{Msq} is minimised to half of the maximum input current, $I_{in_max}/2$, giving a maximum squarer output (from Eq. 7) also equal to $I_{in_max}/2$.

3.2.3 Diffusion network

The diffusion network enables the localised spatial averaging of a register’s contents. For a register storing an image, this functions as an image low pass spatial filtering operation. This is a process that requires significant computation in the digital domain; in the analogue domain, it is a simple, single command giving each PE access to a local regional average. This can be useful, for example, for producing a locally adaptive threshold for object extraction that automatically compensates for differences in object lighting. The additional hardware consists of two N-type transistors configured to act as resistors (with additional switch transistors) between adjacent element’s analogue buses, with diffusion strength controlled by means of an analogue bias voltage. With bus voltage held at a constant voltage by the second phase of the S^2I cell write cycle, variations in inter-PE resistances through v_{gs} change are minimal. Any combination of analogue registers can then act as sources or sinks to the diffusion network (shown in Fig. 6). The network can be locally broken, if required, in the horizontal and/or vertical directions, controlled by outputs from the local digital register bank. The point spread function of a 1-D linear array of PEs assembled as a diffusion network is an exponential function dependent on S^2I cell source resistance, PE interconnect resistance and programming resistance of the S^2I cell [29]. A 2-D resistive grid simulation shows similar properties; however, its analysis is not simply tractable.

The diagram illustrates a 2N1T1R1C1B1C2 architecture. It features a LOCAL READ BUS and a LOCAL WRITE BUS. The array consists of multiple memory cells (Mmem0, Mmem1, etc.) and read/write transistors (R0, R1, etc.). A 'wired OR' connection is shown, linking the LOCAL READ BUS to the LOCAL WRITE BUS via a series of transistors and capacitors. The diagram also shows a Precharge signal and a Vdd supply.

In order to enable conditional execution and global asynchronous processing, register R12 (with R11-controlled load) is exploited. Figure 9 shows the circuit configuration for running asynchronous propagations. Signal R12_out is the storage node of register R12, and R0_out is the storage node of register R0. The arrangement is essentially the concatenation of each PE's R12 output, to the inputs of its neighbour's R12, creating a 2-D logic chain (shown in 1-dimension in the figure). As register R12 of the n -th PE is read, it produces the output at the LRB; this is then fed through an inverter to the analogue/digital bus of the neighbour cell ($n + 1$), writing the data (gated by Global_prop_en, and R0_out) to the LWB, and then, gated through R11_out, onto the storage node of R12. In order to execute the propagation, an initial marker has to be

An array of SIMD PEs executes the same instruction on all processors (PEs) within the array, while data upon which processors operate is local to the PEs. Data can be input to the array via the IO systems of the chip or more usually from light radiation (which naturally is a wholly parallel operation that does not require extensive IO resource). Each PE in the array (in the case of this work) contains 6 general purpose analogue registers which can be likened to signed 8-bit equivalent digital registers. The special purpose analogue register (known as NEWS) that connects a PE to its 4 nearest neighbours, gives it access to neighbour data. Furthermore, each PE has 14-bits of binary storage (R0-R12 and 'flag') available to it. For the

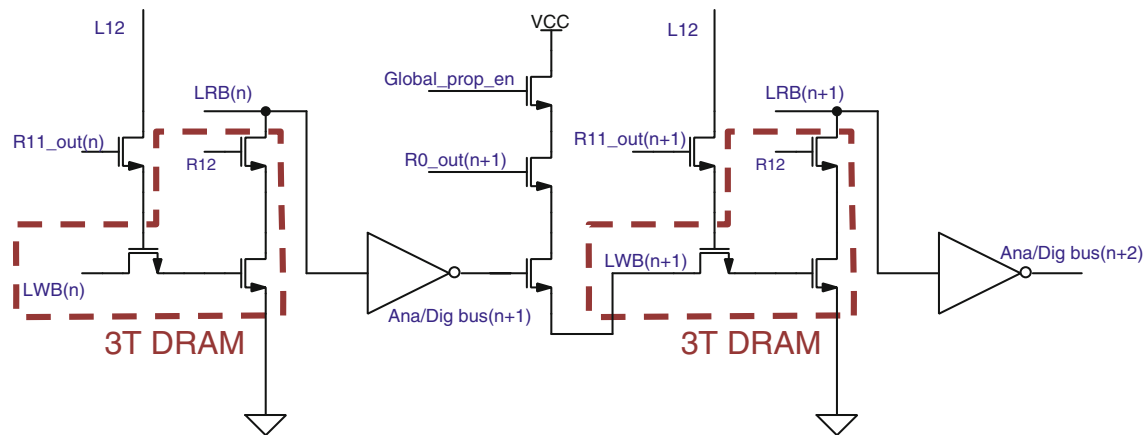


Fig. 9 Memories configured as an asynchronous propagation network

programmer, a register (whether digital or analogue) is always utilized in its entirety across the array, as a planar array structure as depicted in Fig. 10.

Hence an operation of $\mathbf{A} \leftarrow (\mathbf{B} + \mathbf{C})$ (operating in a single instruction cycle) will execute the operation

$$A_{i,j} = -(B_{i,j} + C_{i,j}) \quad (12)$$

for all cells $PE(i,j)$ within the array. As noted above, all analogue operations are inverting in this design.

To facilitate operations with a level of local autonomy, all write operations of analogue registers require that a flag register be “true” for execution to take place. The flag register can be set from logical digital operations, local analogue comparison or external input.

With nearest orthogonal neighbour communication, functions such as edge detection can be enabled. Filters requiring larger neighbourhoods can be easily implemented using multiple neighbour transfers.

The functional example code (running a locally adaptive threshold algorithm) shown in Table 1 illustrates the range of constructs available to the programmer:

5 Implementation and measurement results

The chip was implemented in a 6-metal 1-poly 0.18 μm CMOS technology. The photograph/floorplan of the IC in Fig. 11 has 64×20 pixels in a chip of size

Fig. 10 Array processor view of the SIMD architecture

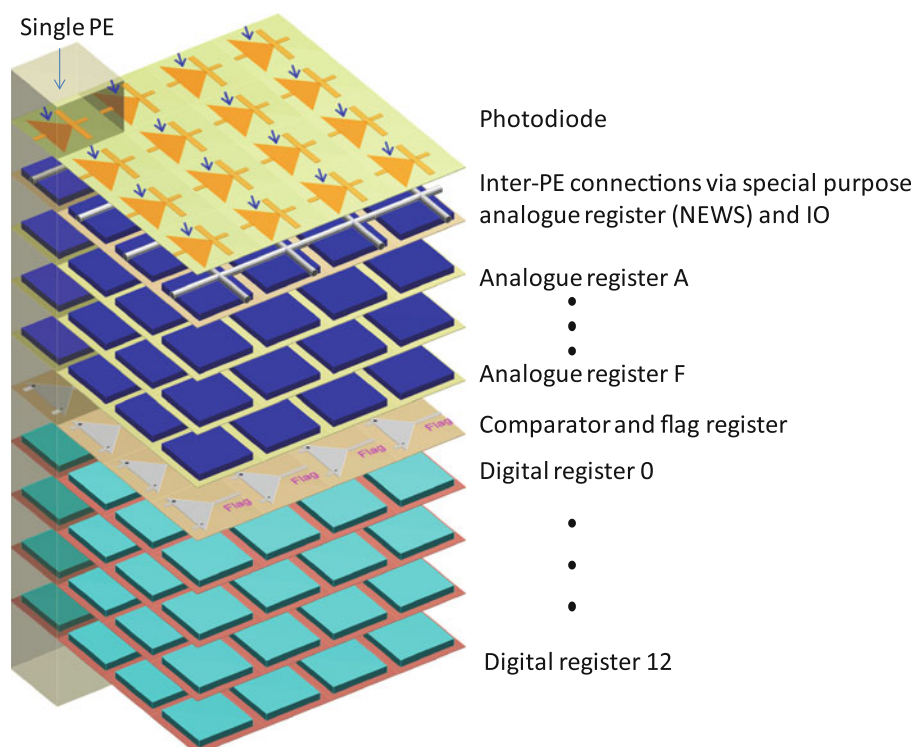
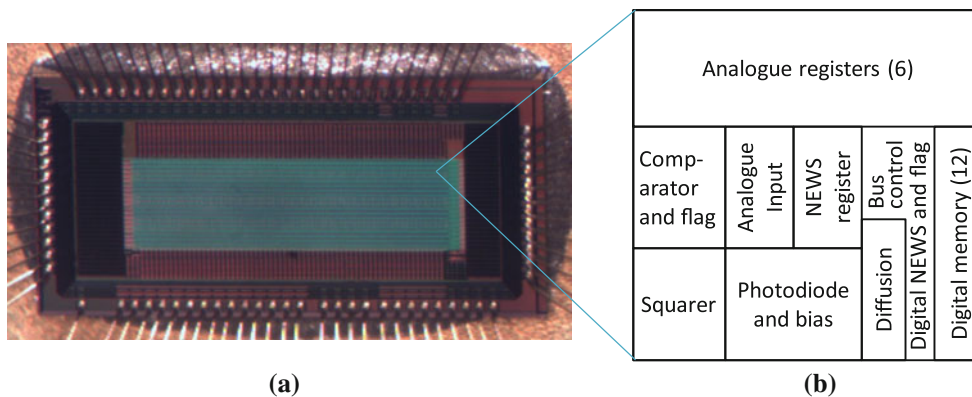


Table 1 SCAMP4 vision chip code for adaptive thresholding

Command	Comment
$A \leftarrow \text{PIX}$	Load analogue register A with inverted pixel value
RESPIX	Reset pixel
$B \leftarrow \text{PIX} + A$	Load analogue register B with inverted sum of pixel value and A; B now holds image data over half of register full-scale-range with dark (no light) represented by zero current. B is a +ve image
$C \leftarrow \text{DIFF}(B,1,1)$	Load analogue register C with image B spatially diffused in x and y directions (controlled by digital switches). C is a -ve image.
$D \leftarrow C + B + \text{IN}(2)$	Add -ve diffused image in C to +ve image B; store inverted result in D. Also a (small) global constant is added here. D is a -ve image
WHERE (D)	Start “Execute in locations where $D > 0$ ” block; sets flag register
$R4 \leftarrow \text{FLAG}$	Load R4 with the flag register value
ALL	Resets flag register i.e. all locations execute following code
OUT B	Outputs image from array B (original image)
OUT C	Outputs inverted diffused image from array C
OUT D	Outputs (image+ inverted diffused image) from array D
OUT R4	Outputs binary thresholded image from R4 (original image with thresholded pixels marked)

3.05×1.525 mm. The PE floorplan contains a pixel with approximate fill factor of 6 %, analogue and digital registers occupy 41 and 15 % of PE space respectively.

The array is designed to operate at 10 MHz with an analogue supply voltage of 1.5 V and digital supply voltage of 1.8 V. The IC has a total quiescent power consumption (i.e. when not executing any instructions) of $31 \mu\text{W}$, and a peak power consumption of $\sim 25 \text{ mW}$.

**Fig. 11** a Chip photograph, with floorplan of PE in (b)

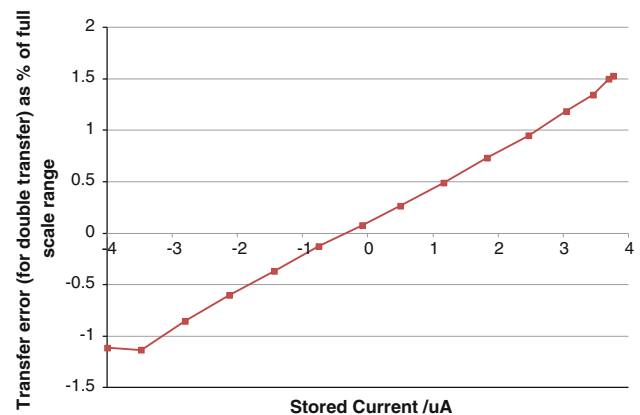
5.1 Analogue performance

Transferring data from one analogue register (**B**) to another (**A**) has errors associated with the operation (unlike its digital counterpart), the most significant generally being the signal dependent error. This error can be simply modelled as:

$$A_{i,j} = -B_{i,j} - \kappa B_{i,j} \quad (13)$$

where κ is a global constant describing this error. For algorithms involving multiple register interchanges, this error term can limit the complexity of implementable algorithms, and is an important parameter in discrete-time analogue processors. Since the native register copy operation is inverting, generally we compare the difference in a register's contents after copy-out and a copy-back operations.

On measurement of this error after the two transfers with the error correction circuit disabled, it was found that the peak-to-peak signal dependent error across the full scale register range (-3.5 to $3.5 \mu\text{A}$) was 2.5 % (see Fig. 12).

**Fig. 12** Register transfer error after two register transfers averaged across array

Using the error correction circuit with a drive potential of 1.4 V, this error could be shifted by 1.8 % of full scale. Unfortunately, a design error deployed a signal of inverted polarity and produced a shift in the direction opposite to that required; for a follow-on chip this will clearly only require a minor design change to provide very low register errors.

An important facet of analogue memories (unlike their digital counterparts) is that a value stored in a register will be different to the value eventually read-out. This is due to the effects of transistor leakage (finite off-state current). It is of particular importance for algorithms requiring motion detection, that are required to store information between video frames, since degradation increases with time. A register can store a current between -4 and $+4$ μA , with the registers always gravitating towards a $+4$ μA stored current irrespective of initial value. Rate of degradation was found to be dependent on stored value as represented in Fig. 13.

Registers decayed at a worst-case rate of 5 % of full-scale range per second (cf. 4 % over 20 ms in [12]), while the vision chip imaged an office scene at 220 lux; with high illumination levels of 70 klux, decay rate increases by a factor of ~ 4 . Over a typical frame interval of 50 ms, this decay rate has a negligible effect.

To test the diffusion network, an image was captured and stored in a register. The diffusion network was then defined in North–South and East–West directions without any inhibiting mask, with minimum resistance paths between PEs. A single register was then copied to another. The diffusion of a letter “U” is shown in Fig. 14; the diffused image is shown inverted for comparison (reversing the inverting effect of register transfer).

The photo-response non-uniformity (PRNU) of the imaging system (as a percentage of full-scale-range) over 80 % of register range, was 1.6 and 1.5 % rms for PE’s containing N-well photodiodes and N+/P-subst photodiodes

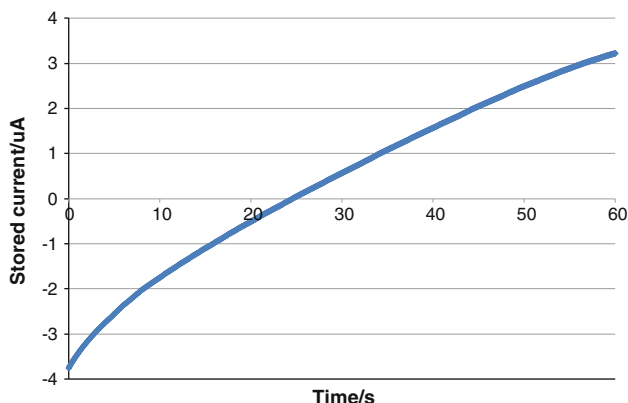


Fig. 13 Average decay of registers' stored current over time from initial stored currents of 3.74 μA (at 21 °C and 220 lux lighting)

respectively. While the N-well photodiode array has slightly higher PRNU, this array provides $3\times$ higher sensitivity than the N+/P-subst photodiodes, through improved responsivity and reduced photodiode capacitance. For the N-well sensors, with the sensor fitted with a F1.4 lens, light sensitivity can be varied by adjustment of Pbias (see Fig. 7) from 140 to 220 LSBs $(\text{Lux.s})^{-1}$ (with 8-bit digitisation). The fill factor is 6.5 % for the N+/P-subst diode and is 5.7 % for the N-well diode.

For evaluation of the squarer circuit, a value stored in the “IN” register was squared and both inputs and outputs recorded when swept over the 1-quadrant input range. The results from the first 10 pixels (for reasons of clarity), along with the ideal response of the system and RMS error are shown in Fig. 15; the error bars indicate the standard deviation of the mean error. Transistor mismatch is a significant contributor to these variations in squarer response.

5.2 Digital performance

Logical operations and asynchronous processing have been found to proceed as designed. Data propagation from cell to cell within the asynchronous network progresses in a time of 16 ns, or uni-directionally across the entire array in 1 μs .

The digital system within the IC only allows the short-term storage of data bits in DRAM; due to leakage of the minimum-sized write transistor, a refresh is required every 50 μs . This is expected to be improved for a successor IC.

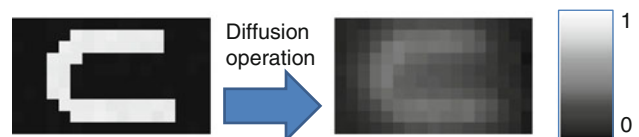


Fig. 14 Spatial diffusion of a letter “U” (single clock cycle operation)

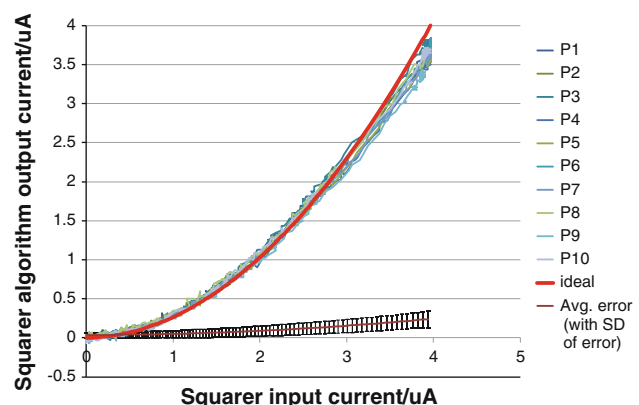


Fig. 15 Operation of squarer for 10 pixels (with ideal response for comparison)

5.3 Code example

The results of the algorithm in Table 1 are shown in Fig. 16. The source figure is an elongated “U” with a varying luminosity of the character, and its background, along its length. A simple binary threshold would result in only partial recovery of the information. With the diffusion network used to create a local average of nearby pixel values, an adaptive threshold is created that can then be applied; the local average (with a global offset) is used to set the threshold that is applied to each individual pixel.

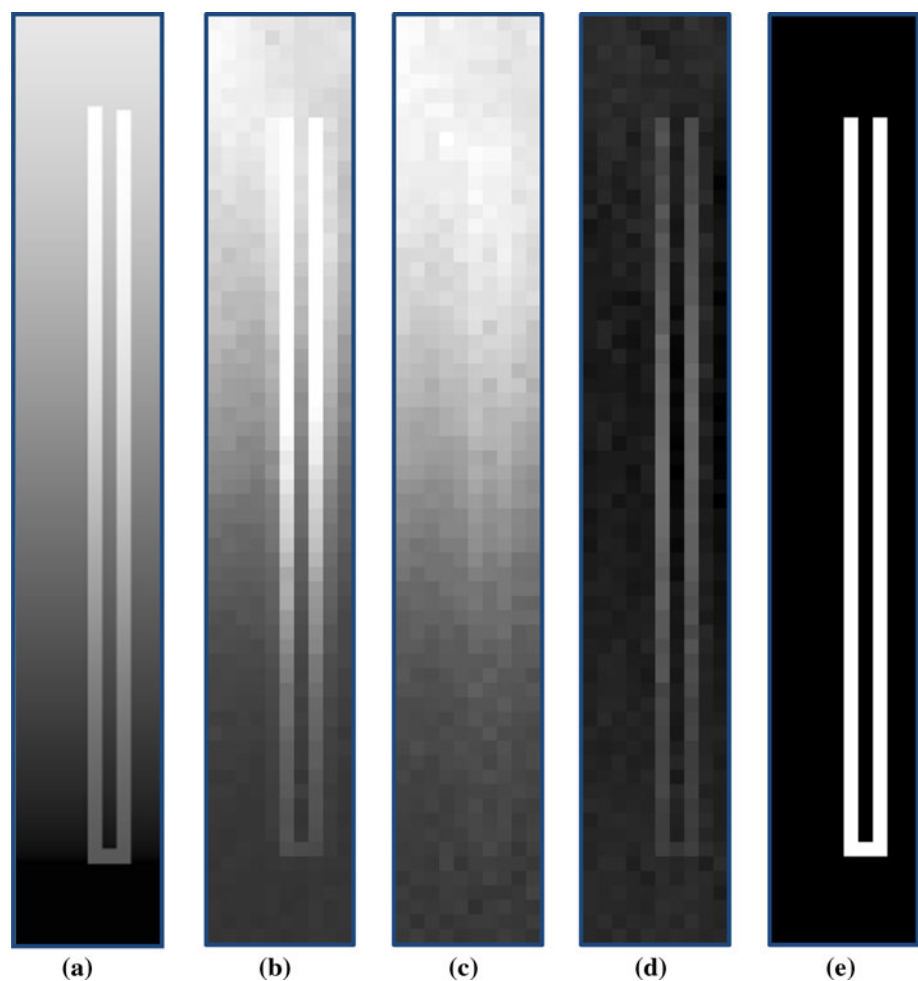
6 Example application

The completed system combines the presented vision chip, an instruction delivery and control system (synthesised upon a Spartan-3 FPGA) and a communications interface to upload algorithms and receive data that the vision chip is programmed to return. Algorithms are developed and compiled using the APRON compiler [30]. Returned data for analysis and inspection is viewed using a custom

viewer program running on a host computer. Unlike many high-speed imaging systems, we wanted to demonstrate that as well as acquiring images at high frame rates, we are also capable of processing them in real-time, on the focal plane, to determine a non-trivial metric. In this example application we wish to discriminate between open and closed binary shapes, i.e. ‘O’ versus ‘U’, outputting from the vision sensor only the processed result. If a closed shape lies within a pre-defined area of the focal-plane, the vision chip system will return ‘true’ to the host, else ‘false’. This occurs at a frame rate of 30,000 fps, limited by the rate instructions can be issued by the system controller

The test apparatus (Fig. 17a) consists of a wheel spinning at high speed. The circumference of the wheel is labelled with three open and three closed shapes, and an extended “filled in” area which is used as a trigger (Fig. 17b). The algorithm consists of two stages. First, as the vision chip starts processing a new series of frames it waits until its entire view is white, then it waits until this is not the case. This transition is used to trigger the second stage of the algorithm which is the discrimination of open and closed shapes. A trigger is necessary to stabilise the visualisation of the

Fig. 16 **a** Input object, **b** acquired image stored in Register B, **c** register C—diffused image, **d** register D—the source register for the thresholding operation—difference between input image and diffused image (with offset), **e** Register R4—threshold result. Images (b) and (c) are shown inverted for clarity



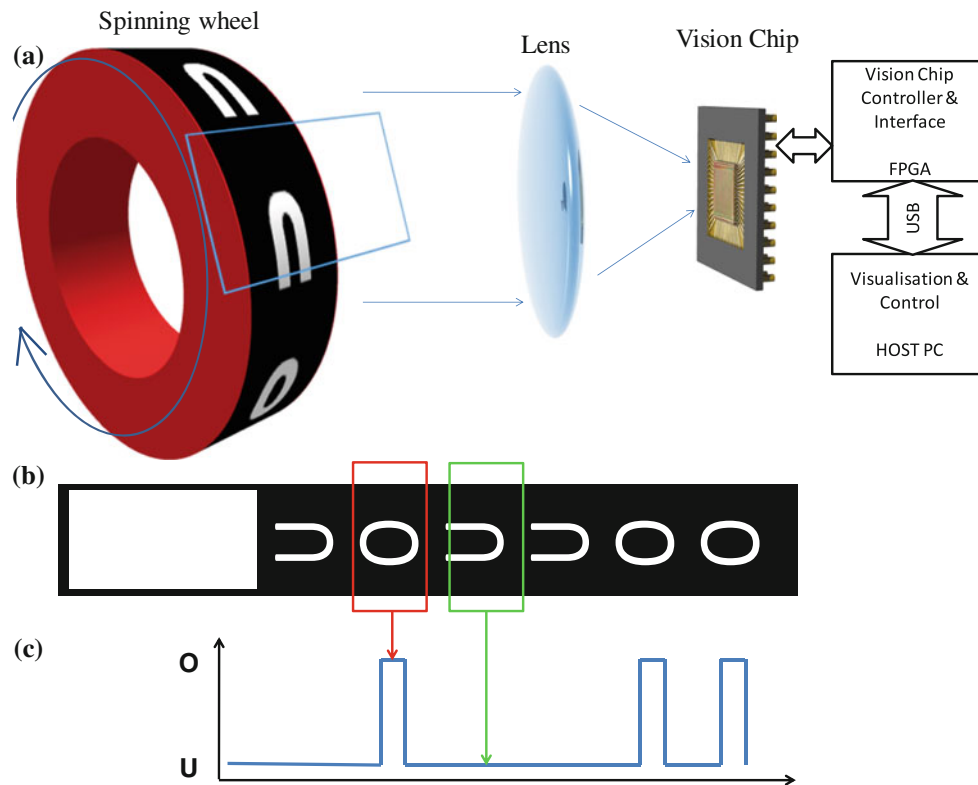
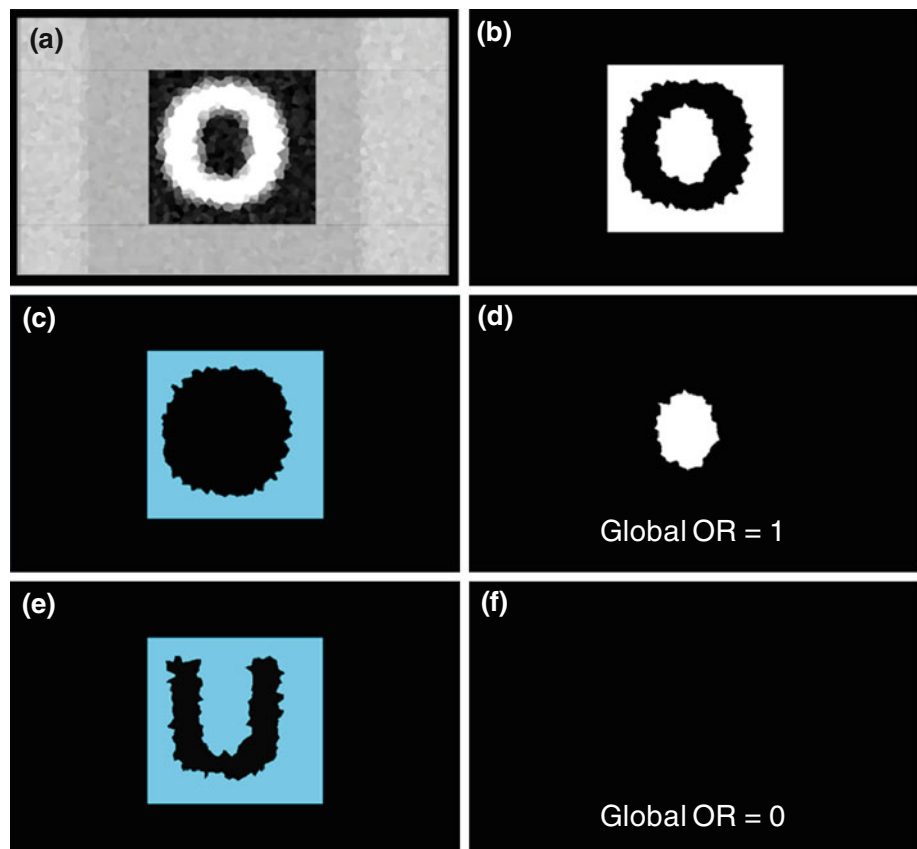


Fig. 17 a Detection of open and closed shapes from a strip (b) with user output in (c)

Fig. 18 Sketch of image processing sequence for distinguishing open and closed shapes. See text for details



returned data (akin to an oscilloscope trace), shown diagrammatically in Fig. 17(c).

The algorithm for detecting open and closed shapes exploits the asynchronous propagation functions of the vision chip and is illustrated in Fig. 18. It consists of the following steps: capture image and define a region of interest around the object by repeatedly shifting in ‘1’ from all 4 boundaries to isolate the region containing the spinning wheel (Fig. 18a); threshold image so background is ‘1’, and shapes are ‘0’, and load the propagation control register R11 (Fig. 18b); runs the asynchronous propagation for an “O” from the perimeter of the box; the propagation wave is inhibited by the shape and does not enter inside the “O”—effectively flood-filling up to where $R11 = '0'$, storing the result in R12 (Fig. 18c); perform array-wide logic operation $R4 = (R12) \cdot R11$ (Fig. 18d). A global readout operation is then performed, where the output result is the OR operation on all elements of R4. If an object is closed, the flood fill propagation does not enter the ‘hole’ and the returned result is true; if the shape is open as shown in Figs. 18(e), (f), the global OR result is false.

The wheel on average spins at 3,000 rpm; with a radius of 23 mm, the shapes move at 7.2 m/s. After triggering, the vision chip samples images, at 30,000 fps, returning the one bit of resultant data per frame (indicating a presence of a closed shape in a frame). To enhance the visualisation, the user can select a single sampling point at which an entire single frame is returned and thus can visually inspect if the frame contains an open or closed shape, and whether the waveform is in agreement. Since carrying out this work, we have demonstrated closed object detection at 100 kfps within a 256×256 array [6].

7 Conclusions

A prototype 20×64 vision sensor/processor chip has been tested and successfully applied to an application in high speed imaging operating at 30,000 fps. This has demonstrated the utility of asynchronous propagation and data reduction of on-focal plane processing, and provided the post-silicon validation of the new PE design. The follow-up vision chip implementation, scaled-up to a larger array size [31], will enable high-speed and efficient implementation of image processing algorithms in practical machine vision applications.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Gu, Q., Aoyama, T., Takaki, T., & Ishii, I. (2013). High frame-rate tracking of multiple color-patterned objects. *Journal of Real-Time Image Processing*. doi:10.1007/s11554-013-0349-y.
- de Albuquerque, M. P., Chacon, G. T., de Faria, E. L., & Murari, A. (2012). High-speed image processing algorithms for real-time detection of MARFES on JET. *IEEE Transactions on Plasma Science*, 40(12), 3485–3492.
- Goda, K., Ayazi, A., Gossett, D. R., Sadasivam, J., Lonappan, C. K., Sollier, E., et al. (2012). High-throughput single-microparticle imaging flow analyzer. *Proceedings of the National Academy of Sciences*, 109, 11630.
- Blug, A., Strohm, P., Carl, D., Hofler, H., Blug, B., & Kailer, A. (2012). On the potential of current CNN cameras for industrial surface inspection. *2012 13th international workshop on cellular nanoscale networks and their applications (CNNA)*, IEEE.
- Lahdenoja, O., Sääntti, T., Poikonen, J., Laiho, M., & Paasio, A. (2013). Characterizing spatters in laser welding of thick steel using motion flow analysis. *Image analysis* (pp. 675–686). Berlin, Heidelberg: Springer.
- Carey, S. J., Barr, D. R. W., Wang, B., Lopich, A., & Dudek, P. (2012). Locating high speed multiple objects using a SCAMP-5 Vision-Chip. *IEEE workshop on cellular nanoscale networks and applications, CNNA 2012*, Turin.
- Carey, S. J., Barr, D. R. W., & Dudek, P., Low power high-performance smart camera system based on SCAMP vision sensor. *Journal of Systems Architecture* (in press).
- Zhang, W., Fu, Q., & Wu, N. (2011). A programmable vision chip based on multiple levels of parallel processors. *IEEE Journal of Solid-State Circuits*, 46, 2132–2147.
- Komuro, T., Kagami, S., & Ishikawa, M. (2004). A dynamically reconfigurable SIMD processor for a vision chip. *IEEE Journal of Solid-State Circuits*, 39, 265–268.
- Lopich, A., & Dudek, P. (2010). An 80×80 general-purpose digital vision chip in $0.18 \mu\text{m}$ CMOS technology. *IEEE International Symposium on Circuits and Systems, ISCAS 2010* (pp. 4257–4260).
- Dudek, P., & Carey, S. J. (2006). A general-purpose 128×128 SIMD processor array with integrated image sensor. *Electronics Letters*, 42(12), 678–679.
- Ginhac, D., Dubois, J., Paindavoine, M., & Heyrman, B. (2008). An SIMD programmable vision chip with high-speed focal plane image processing. *EURASIP Journal of Embedded Systems*, 2008, 1–13.
- Fernandez-Berni, J., Carmona-Galán, R., & Carranza-González, L. (2011). FLIP-Q: A QCIF resolution focal-plane array for low-power image processing. *IEEE Journal of Solid-State Circuits*, 46, 669–680.
- Cottini, N., Gottardi, M., Massari, N., Passerone, R., & Smilansky, Z. (2013). A $33 \mu\text{W}$ 64×64 pixel vision sensor embedding robust dynamic background subtraction for event detection and scene interpretation. *IEEE Journal of Solid-State Circuits*, 48(3), 850–863.
- Rodriguez-Vazquez, A., Dominguez-Castro, R., Jimenez-Garrido, F., Morillas, S., Garcia, A., Utrera, C., et al. (2010). A CMOS vision system on-chip with multi-core, cellular sensory-processing front-end. In C. Baatar, W. Porod, & T. Roska (Eds.), *Cellular nanoscale sensory wave computing* (p. 129). New York: Springer.
- Poikonen, J., Laiho, M., & Paasio, A. (2009). MIPA4k: A 64×64 cell mixed-mode image processor array. *IEEE international symposium on circuits and systems, ISCAS 2009* (pp. 1927–1930).
- Gasparini, L., Manduchi, R., & Gottardi, M. (2010). An ultra-low-power contrast-based integrated camera node and its application as a people counter. In *Seventh IEEE international conference on advanced video and signal based surveillance (AVSS)* (pp. 547–554).
- Hoffmann, R., Weikersdorfer, D., & Conradt, J. (2013). Autonomous indoor exploration with an event-based visual SLAM system. European conference on mobile robots, Barcelona, Spain.

19. Carey, S. J., Barr, D. R. W., Wang, B., Lopich, A., & Dudek, P. (2012). Mixed signal SIMD cellular processor array vision chip operating at 30,000 fps. In *19th IEEE international conference on electronics, circuits and systems (ICECS)* (pp. 324–327).
20. Zarandy, A., Dominguez-Castro, R., & Espejo, S. (2002). Ultra-high frame rate focal plane image sensor and processor. *IEEE Sensors Journal*, 2, 559–565.
21. Carey, S. J., Lopich, A., & Dudek, P. (2011). A processor element for a mixed signal cellular processor array vision chip. *IEEE international symposium on circuits and systems, ISCAS 2011*, Rio de Janeiro.
22. Dudek, P., & Hicks, P. J. (2000). A CMOS general-purpose sampled-data analogue processing element. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(5), 467–473.
23. Dudek, P. (2006). An asynchronous cellular logic network for trigger-wave image processing on fine-grain massively parallel arrays. *IEEE Transactions on Circuits and Systems II*, 53(5), 354–358.
24. Dudek, P. (2000). *A programmable focal-plane analogue processor array*. Ph.D. thesis, University of Manchester Institute of Science and Technology (UMIST).
25. Dudek, P. (2011). SCAMP-3: A vision chip with SIMD current-mode analogue processor array. *Focal-plane sensor-processor chips* (pp. 17–43). New York: Springer.
26. Toumazou, C., Hughes, J. B., & Battersby, N. C. (Eds.). (1993). *Switched-currents: An analogue technique for digital technology*. London: Peter Peregrinus Ltd.
27. Huang, C. Y., Chen, C. Y., & Liu, B. D. (1995). Current-mode linguistic hedge circuit for adaptive fuzzy logic controllers. *Electronics Letters*, 31(17), 1517–1518.
28. Hughes, J. B., & Moulding, K. W. (1993). S2I: a switched-current technique for high performance. *Electronics Letters*, 29, 1400–1401.
29. Shi, B. E., & Chua, L. O. (1992). Resistive grid image filtering: input/output analysis via the CNN framework. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 39(7), 531–548.
30. Barr, D. R. W., & Dudek, P. (2009). APRON: a cellular processor array simulation and hardware design tool. *EURASIP Journal on Advances in Signal Process*, 2009, 1–9.
31. Carey, S. J., Barr, D. R. W., Lopich, A., & Dudek, P. (2013). A 100,000 fps Vision Sensor with Embedded 535 GOPS/W 256x256 SIMD Processor Array. *VLSI circuits symposium*, Kyoto, Japan, 12–14 June 2013 (accepted).



Stephen J. Carey is a Research Fellow, with over 15 years experience in electronics and optoelectronics, including analogue and mixed-mode IC design, and system design for sensor interfaces. He received the B.Eng. degree in Electrical and Electronic Engineering from the University of Bristol, UK in 1987, followed by an M.Sc. in Microwaves and Modern Optics from University College London, UK in 1989. In 2001, he received a Ph.D. in

optical tomography from UMIST, UK. He joined Sifam Ltd in UK in 1989 and then Bookham Technology in 1996 working on near infrared optical fibre components. In 1998 he joined UMIST (now University of Manchester) developing a near infra-red absorption tomography system. From 2001, he has been developing systems for

vision chips and high frequency ultrasound transducers. The latter work requiring a mix of high voltage and low noise electronics produced a prototype of one of the world's highest resolution phased array ultrasound scanners designed specifically for skin cancer detection. He has been working entirely on vision chips and their systems since August 2009.



David R. W. Barr is a Research Associate in the School of Electrical and Electronic Engineering at the University of Manchester whose research contributions include the design of embedded control systems, driver software, compilation and rapid application development tools for processor array devices. He is the primary developer of APRON, a software package for the modelling and development of a variety of cellular processor array devices. He achieved his M.Eng. in Computer Systems Engineering at the University of Manchester Institute for Science and Technology (UMIST) in 2005, and a Ph.D. at the University of Manchester in 2009. His M.Eng. dissertation was on the design and construction of a prototype vision chip control system, which he continued as a research assistant after graduation, resulting in the first generation SCAMP3 system. His Ph.D. thesis explored the application of parallel processors and processor arrays to large neural-network simulations, image processing algorithms and embedded computer vision.



arrays.

Bin Wang received his B.Eng. degree in Electronic Information engineering from Xi'an University of Technology, China, in 2008, and his M.Sc. degree in Data Communications from the University of Sheffield, UK, in 2009. He is currently a Ph.D. Candidate at the University of Manchester. His research interests include image processing, computer vision, and Statistical Machine Learning, especially parallel image processing algorithms for processor



Alexey Lopich received the M.Eng. degree in mathematical electronics from the Faculty of Mathematics and Mechanics at Belarusian State University, Minsk, Belarus in 2003 and Ph.D. degree from The University of Manchester, Manchester, UK in 2007. He has worked as a microelectronic engineer in 2003 and has been working as a Research Associate at The University of Manchester since 2006. His research interests are in image processing architectures and algorithms, VLSI design, smart sensors, cellular arrays, machine vision and embedded systems.



Piotr Dudek received his mgr inż. degree from the Technical University of Gdańsk, Poland, in 1997, and the M.Sc. and Ph.D. degrees from the University of Manchester Institute of Science and Technology (UMIST) in 1996 and 2000 respectively. He worked as a Research Associate, and since 2002 as a Lecturer, at The University of Manchester. During 2008/09 he was a Visiting Associate Professor in the Department of Electronic and

Technology. He is currently a Reader in the School of Electrical and Electronic Engineering, The University of Manchester. His research interests are in the area of integrated circuit design, cellular processor arrays, vision sensors, novel computer architectures and brain-inspired systems.

Computer Engineering at the Hong Kong University of Science and